Average Complexity Analysis of Scalar Quantizer design

[1]JELENA NIKOLIC, [1]ZORAN PERIC, [2]DRAGOLJUB POKRAJAC
[1]Telecommunication Department, [2]Computer and Information Sciences Department and Applied Mathematics Research Center
[1]Faculty of Electronic Engineering Nis, [2] Delaware State University
[1]Aleksandra Medvedeva 14, 18000 Nis, [2]1200 N DuPont Hwy, Dover, Delaware 19901
[1]SERBIA, [2]USA

Abstract: **-** In this paper an exact and complete analysis of average design complexities of Lloyd-Max's scalar quantizers, scalar compandors and scalar quantizers designed using the hybrid model is carried out. The average design complexity depends on arithmetic complexity, memory complexity and implementation complexity. It is demonstrated that for a fixed number of quantizaton levels *N*, scalar compandors have the smallest and the Lloyd-Max's scalar quantizers have the largest complexity. Furthermore, it is shown that for a fixed number of quantizaton levels *N* the average design complexity of hybrid scalar quantizers is significantly smaller than the average design complexity of Lloyd-Max's scalar quantizers. Combining this result with the fact that the performances of hybrid scalar quantizers are almost equal to the optimal performances of Lloyd-Max's scalar quantizers, the usability of recently developed hybrid model is confirmed.

Keywords: Average design complexity, Compandor, Hybrid quantizer, Lloyd-Max's quantizer.

1 Introduction

Quantization is the process of replacing analog samples with approximate values taken from set of allowed values [1, 2, 3]. An *N*-level scalar quantizer can be defined as a functional mapping of the set of real numbers *R* onto the set of the output representation. The set of the output representations, called the representation levels $\{y_1, y_2, \ldots, y_N\}$ constitutes the code book that has the size $|C|=N$. Associated with every *N*-level scalar quantizer is partition of the set of real numbers *R* into *N* cells $R_i==(x_{i-1}, x_i]$, $i=1,\ldots,N$, where x_i $i=0,1,\ldots,N$ are decision thresholds. Therefore, a quantizer can be uniquely determined by its partition and the set of the output representation. Hence, design of scalar quantizer is equivalent to selection of the representation levels and the partition or cells for a fixed number of quantization levels *N*. Every quantizer can be viewed as the combination of encoder and decoder [4]. Encoder transmits the index *i* of the selected representation level y_i, assigned to an input sample. Decoder reconstructs the corresponding reproduction value by using table-lookup procedure [4].

There are several models of scalar quantizers that are based on different quantization techniques. The principal goal of scalar quantizer design is to decide which model will be used while designing as well as to select the encoder and decoder that provide the best possible reproduction of the original signal attaining the smallest possible design complexity. However, to determine a quantizer design complexity is not a straightforward task. Until now there has not been much theoretical or even quantitive comparison among the design complexities of scalar quantizers that are based on different models. Consequently, much work is still need in order to determine which model provides the best performances versus complexity trade off and to gain an understanding why certain complexity-reducing models are better than others.

In this paper, we propose and perform an exact and complete analysis of the average design complexities of scalar quantizers based on different models. We define the average design complexity as the arithmetic mean of the arithmetic complexity, implementation complexity and memory complexity. Note that one of the reasons of carrying out the analysis of average design complexities is to determine if the recently developed hybrid model of scalar quantizers [1, 2] reduces the average design complexity of Lloyd-Max's quantizers. Hence, using the proposed definition of the complexity, we performed and demonstrated the quantitative comparison among average design complexities of compandors, hybrid quantizers and Lloyd-Max's quantizers when the number of quantization levels varies *N*=32, 64, 128, 256, 512, 1024.

2. Average design complexity

The average design complexity of scalar quantizer is usually specified as arithmetic mean of the arithmetic complexity, memory complexity and implementation complexity [3]. Arithmetic (or computational) complexity is defined as the number of arithmetic operations per sample that must be performed when encoding or decoding. Memory (or storage) complexity is defined as amount of auxiliary storage or memory space that is required to store the parameters that specify the considered scalar quantizer model (for encoding and decoding). Implementation complexity is defined as the number of elementary digital blocks (circuits) that are needed for scalar quantizer construction.

Let us denote the aforementioned complexities with: A (arithmetic complexity), M (memory complexity), I (implementation complexity). The average design complexity can be defined with [3]:

$$K = w_1 A + w_2 M + w_3 I \qquad (1)$$

where w_1, w_2, w_3 are the weights of arithmetic complexity, memory complexity and implementation complexity. Assuming that all weights are equal, average design complexity is defined as arithmetic mean of the arithmetic complexity, memory complexity and implementation complexity:

$$K = \frac{1}{3}(A + M + I). \qquad (2)$$

Using Eq. (2) we perform the analysis when the quantizer model corresponds to compandor, hybrid quantizer and the Lloyd-Max's quantizer. It is very important to point out that in the following analysis we consider the equality of arithmetic and implementation complexities. This equality can be assumed in case of scalar quantizers that are modeled by using elementary digital blocks which are called primitive because of performing standard (elementary) functions. In such a case it is obvious that arithmetic and implementation complexities are defined with the number of primitive blocks, while memory complexity is defined with the number of parameters that describe scalar quantizer model.

2.1. Analysis of Lloyd-Max's quantizer average design complexity

Lloyd and Max proposed an algorithm to compute optimum quantizers using the mean-square error distortion measure [4, 5, 6, 7]. The analysis of Lloyd-Max's quantizer average design comlexity begins by considering the Lloyd-Max's quantizer implementation that is shown in Fig. 1. The implementation of the Lloyd-Max's quantizer

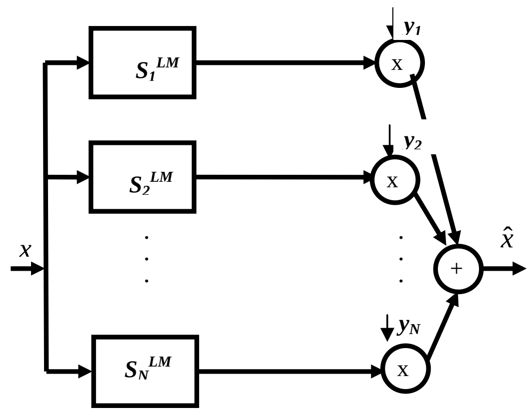

Fig. 1. Implementation of Lloyd-Max's quantizer

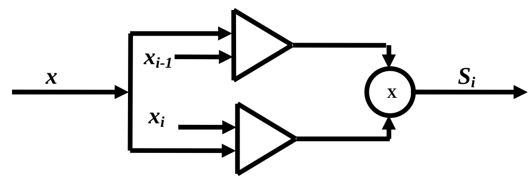

Fig. 2. Implementation of selector with primitive blocks (two comparators and one multiplier)

consists of blocks indicated with S_1^{LM}, S_2^{LM},..., S_N^{LM} that represent selector functions, defined by $S_i^{LM}(x)$=1 if $x \in R_i$ and 0 otherwise, and the multipliers indicated by circles that have weight values given by the corresponding output representation y_1, y_2,..., y_N. From the Fig.1. it is obvious that the operation of quantization can be expressed in the following form [4]:

$$O(x) = \hat{x} = \sum_{i=1}^{N} y_i S_i(x). \qquad (3)$$

It is very important to point out that the selector operation, although a relatively simple building block, is not primitive since it can be further decomposed into elementary comparators. Thus, the selector functions can be implemented with two comparators, addition or multiplicaton, as depicted in Fig. 2., or using logical AND device [4]. Considering the fact that each of N selector functions can be replaced by three primitive blocks we derive the following expression for the arithmetic complexity of Lloyd-Max's scalar quantizers:

$$A^{LM} = 3N(selectors) + N(multipliers) + 1(adder)$$

$$A^{LM} = 4N + 1. \qquad (4)$$

Since the arithmetic complexity is considered equal to implementation complexity we can use Eq. (4) to calculate the implementation complexity.

In order to define the expression for memory complexity it is necessary to define the set of scalar quantizer parameters when considering particular model of scalar quantizer. The set of Lloyd-Max's quantizer parameters consists of N representation levels and N+1 decission thresholds which define the memory complexity of Lloyd-Max's scalar quantizers denoted with M^{LM}:

$$M^{LM} = N(representation \quad levels) + \\ + N + 1(decission \quad thresholds) = 2N + 1. \quad (5)$$

Combining Eqs. (2), (4) and (5) we can derive the expression that provides quantitive evaluation of average designing complexity in case of Lloyd-Max's scalar quantizers, denoted with K^{LM}:

$$K^{LM} = \frac{1}{3}(10N + 3). \quad (6)$$

2.2 Analysis of scalar compandor average design complexity

The structure of a nonuniform quantizer, depicted in Fig. 3., consisting of a compressor C, a uniform quantizer, and expandor E in cascade, is called the compandor [4,8,9]. Hence, the arithmetic complexity of a scalar compandor is determined by the arithmetic complexities of compressor, uniform quantizer, and expandor. An uniform quantizer, consisting of encoder and decoder denoted with E^k and D^k respectively, has the arithmetic complexity equal to complexity of Lloyd-Max's scalar quantizers with the same number of quantization levels N since uniform quantizer can be implemented using the same building blocks as those of Lloyd-Max's scalar quantizers. Thereby, the arithmetic complexity of scalar compandor can be specified with following expression:

$$A^k = 4N + 1(uniform \quad quantizer) + 1(compressor) \\ + 1(\exp andor) = 4N + 3. \quad (7)$$

In the last equation we consider the fact that any compressor and expandor can be viewed as nonlinear amplifiers that performs one simple operation [4].

It is important to emphasize that a uniform quantizer is completely specified by support region of the quantizer, i.e. by maximal input signal magnitude that can be fed to the quantizer not resulting in quantizer overload, and by the number of quantization levels N [4]. Therefore, it is obvious that memory complexity of uniform quantizer has a constant value (two). By reserving two memory addresses in order to describe the compressor and expandor functions each we completely specified memory complexity of compandor. Therefore, we can derive the following expression for evaluating the memory complexity of scalar compandors, denoted with M^k :

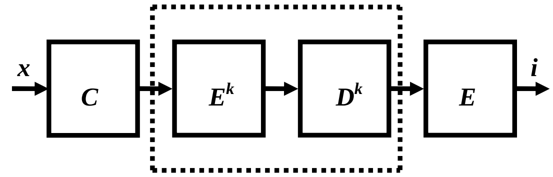

Fig. 3. Implementation of compandor

$$M^k = 2(uniform \quad quantizer) + \\ + 2(compressor \quad and \quad \exp andor) = 4. \quad (8)$$

Analyzing the memory complexity of scalar compandors we deduce that it has very small and constant value not depending on the number of quantization levels. Therefore we confirm that the use of the compandor model can be very helpful when designing quantizers with a large number of quantization levels [8].

Assuming that the arithmetic complexity is equal to the implementation complexity, average design complexity of the stalar compandor can be carried out combining Eqs. (2), (7), (8):

$$K^k = \frac{1}{3}(8N + 10). \quad (9)$$

2.3. Analysis of hybrid quantizer average design complexity

In order to reduce the deficiencies introduced by Lloyd-Max's quantizer and compandor that were observed and analyzed in [1, 2, 9], we proposed a new model of scalar quantizer. The proposed model of scalar quantizer denoted as the hybrid model combines the Lloyd-Max's quantizer model and the compandor model. Namely, for L<<N, applying the model of compandor to inner region O_2 (depicted in Fig. 4), i.e., to N-2L inner cells, and the Lloyd-Max's model to outer region that is union of regions O_1 and O_3 (also depicted in Fig. 4), i.e., to 2L outer cells, it is possible to design the N-levels hybrid scalar quantizer. Observe that the hybrid model is a general quantization model which for L=N/2 represents the Lloyd-Max's model, while in case of L=0 represents the model of compandor.

Let us consider the hybrid model implementation. From the Fig. 5. it is evident that the considered input signal x, is brought first to the inputs of selectors denoted with S_1, S_2 and S_3 which functions are defined as S_i(x)=1 if x∈ O_i and 0 otherwise, i=1, 2, 3 [4]. Also, the considered input signal x is brought to the switches, denoted with SW_1, SW_2, and SW_3. Depending on the region in which the input signal lies, one of the aforementioned switches closes, enabling the input signal to be passed to the one of three possible branches:

Branch 1. to the encoder 1, denoted with E_1
Branch 2. to the compressor, denoted with C and than to the encoder 2, denoted with E_2
Branch 3. to the encoder 3, denoted with E_3.
The adequacy of introducing the structural blocks, denoted with E_1, E_2, E_3, D_1, D_2, and D_3, results from the fact that any scalar quantizer can be viewed as the combined effect of two operations, performed by encoder and decoder.

Note that the processes applied on the signal, transmitted along the first and the third branches are equal. First, let us consider the case when the input signal is brought to the encoder E_1. In such a case, the hybrid model proposes the use of Lloyd-Max's model, i.e., the use of optimal encoder and decoder (E_1 and D_1). Encoder E_1 generates index that is then transmitted over the communication channel. At the received end of the communication system, the transmitted index is brouht to the input of three blocks, denoted with Sd_1, Sd_2 i Sd_3, known as decoder selectors and defined with:
$Sd_1(x)=1$ if $x \in I_1=[1,L]$ and 0 otherwise;
$Sd_2(x)=1$ if $x \in I_2=[L+1,N-L-1]$ and 0 otherwise;
$Sd_3(x)=1$ if $x \in I_3=[N-L, N]$ and 0 otherwise.
For instance, when Sd_1 is indicating that index i belong to the range I_1, switch, denoted with SW_4 is closing, enabling the index to be sent to the decoder D_1 which is subsequently generating the output representation.

Let us now consider the case when the input signal is brought to the compressor. In such a case, the hybrid model proposes the use of the compandor model, i.e., the use of compressor (C), uniform quantizer and expandor (Ex). Note that the uniform quantizer can be represented as a combination of encoder E_2 and decoder D_2. If the decoder selector Sd_2 allows closedown of SW_5, the index generated by E_2 and transmitted over the channel is brought to the decoder D_2. In order to obtain the output representation, the output of the observed decoder needs to be expanded by the expandor. Herewith, we accomplish the detail explanation of the hybrid model implementation.

Let us consider the complexities of a hybrid quantizer. From Fig.5 it is evident that the arithmetic and implementation complexities are determined with the arithmetic complexity of Lloyd-Max's

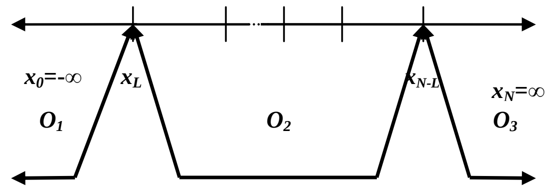

Fig. 4. Illustration of the inner region and the outer region of the hybrid scalar quantizer

quantizer having L quantization levels, as well as with the arithmetic complexity of compandor having N-2L quantization levels and the combination logic that consists of selectors S_1, S_2, S_3, Sd_1, Sd_2, Sd_3, switches SW_1, SW_2, SW_3 SW_4, SW_5, SW_6, and two adders. Therefore, the arithmetic complexity of hybrid quantizer, denoted with A^H, is given as:

$$A^H = 6\times 3(selectors) + 6(switches) + 2(adders) + \\ + 2\times(4L+1)(Lloyd \quad Max \quad quantizer) + \\ + (4(N-L)+3)(compandor) = 4N + 4L + 31. \quad (10)$$

The memory complexity of hybrid scalar quantizer is also determined by the memory complexities of Lloyd-Max's quantizer and compandor. Considering the fact that Lloyd-Max's quantizer, applied when designing outer region of hybrid quantizer, can be completely defined with the threshold between the inner and the outer region, denoted in the Fig. 4. with $x_{N-L}=-x_L$ and with the 2L reconstruction offsets (distances from the representation levels to the nether decision thresholds) [1, 2], memory complexity of a hybrid quantizer can be given as:

$$M^H = 2L(reconstruction \quad offsets) + 4(compandor)$$

$$M^H = 2L + 4 \ . \quad (11)$$

Finally, the average design complexity of the hybrid quantizer is:

$$K^H = \frac{1}{3}(8N + 10L + 35) \ . \quad (12)$$

3. Numerical results

Analyzing the average designing complexities of Lloyd-Max's scalar quantizers, scalar compandors and hybrid scalar quantizers we ascertain that these complexities grows with the number of quantization levels N. Numerical values of the aforementioned complexities are computed using the Eqs. (6), (9) and (12) when the number of quantization levels varies N=32, 64, 128, 256, 512, 1024 and are listed in the Table 1. Observe that analysis of the performances of the hybrid model [1, 2] indicates the optimal value of L=2. Therefore, the analysis is

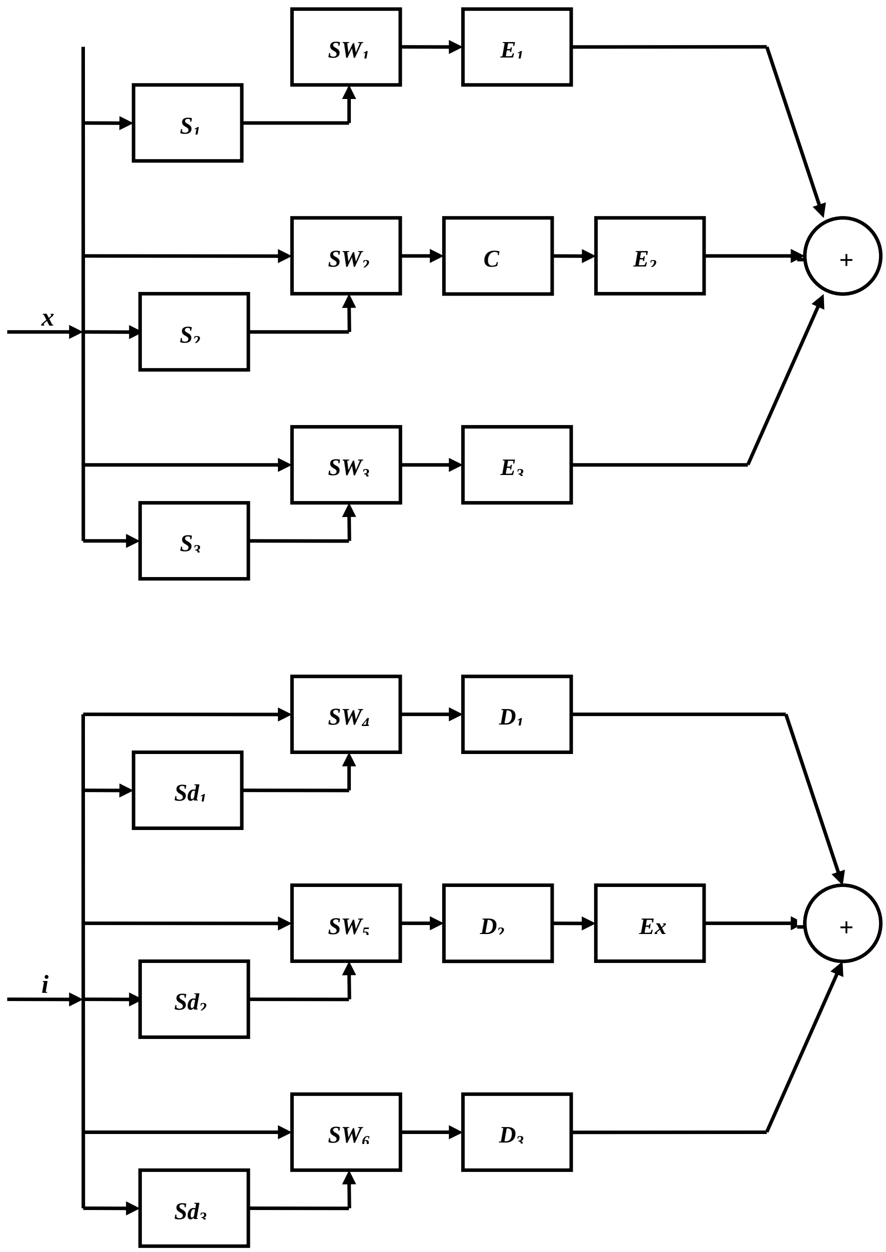

Fig. 5. Implementation of the hybrid scalar quantizer model;

carried out assuming the specified parameter value of *L*.

Results from Table 1 indicate that the smallest average design complexity, for a fixed number of quantization levels *N*, corresponds to scalar compandors while the largest average design complexity corresponds to Lloyd-Max's scalar quantizers. Also, it is obvious that the average design complexity of hybrid scalar quantizers, in case of large number of quantization levels *N* (*N*=256, 512, 1024), is significantly smaller than the average designing complexity of the Lloyd-Max's scalar quantizers.

	K^{LM}	K^{K}	K^{H} (L=2)
***N*=32**	107.67	88.67	103.67
***N*=64**	214.33	174	189
***N*=128**	427.67	344.67	359.67
***N*=256**	854.33	686	701
***N*=512**	1707.67	1368.67	1383.67
***N*=1024**	3414.33	2734	2749

Table 1. Comparison of average design complexities

4. Conclusion

Numerical results demonstrate that the smallest average design complexity, for a fixed number of quantization levels N, corresponds to scalar compandors while the largest complexity corresponds to the Lloyd-Max's scalar quantizers. The memory complexity of scalar compandors is small and constant, not depending on the number of quantization levels. Although we have already shown that the performances of compandor are far from optimal [1, 2], here we demonstrated that the use of compandor model can still be very helpful for designing quantizers with a large number of quantization levels when the small design complexity is the primary engineers' goal. However, when the task is to approach the optimal performance, we propose the use of hybrid scalar quantizer which can provide acceptable compromise between the optimal performances and the design complexity. Specifically, it is already shown [1,2] that the performances of hybrid scalar quantizers are almost equal to the optimal performances of Lloyd-Max's scalar quantizers. In turn, here we have demonstrated that in case of large number of quantization levels N (N=256, 512, 1024) by using the hybrid scalar quantizers the average design complexity is significantly reduced in comparison to the average design complexity of the Lloyd-Max's scalar quantizers. Therefore the analysis presented in this paper has the practical importance. Particularly, it provides justification for near-optimal and complexity-reducing design strategies of scalar quantizers having large number of quantization levels N (N=256, 512, 1024) that can be used for efficient source coding of images [4] and speech [9].

5. Acknowledgement

D. Pokrajac has been partially supported by NIH-funded Delaware IDeA Network of Biomedical Research Excellence (INBRE) Grant, DoD HBCU/MI Infrastructure Support Program (45395-MA-ISP Department of Army), and by National Science Foundation (NSF) Grants (awards # 0320991, #0630388, #HRD-0310163).

References

[1] Zoran Peric, Jelena Nikolic and Dragoljub Pokrajac, Hybrid Scalar Quantizer for the Laplacian Source, *WSEAS Transactions on Communications*, Vol. 6, No 1, pp. 60-65, January, 2007.

[2] Zoran Peric, Jelena Nikolic and Dragoljub Pokrajac, Optimal designing scalar quantizers using a hybrid quantization method for the Laplacian source, *Proceedings of the 4th WSEAS International Conference on Electromagnetics*, *Wireless and Optical Communications*, pp. 156-162, Venice, Italy, November 20-22, 2006.

[3] Gray M. Robert, Neuhoff L. David, Quantization, *IEEE Transactions on Information Theory*, Vol. 44, No. 6, pp. 2325-2383, October, 1998.

[4] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer, Academ. Pub, 1992, Chapter 6, pp. 173-202.

[5] S. P. Lloyd, Least squares quantization in PCM, unpublished memo., Bell Lab.,1957; *IEEE Transactions on Information Theory*, Vol. IT-28, Mar. 1982, pp. 129-137.

[6] J. Max, Quantizing for minimum distortion, *IRE, Transactions on Information Theory*, Vol. IT-6, Mar. 1960., pp. 7-12.

[7] Neil Judell and Louis Scharf, A Simple Derivation of Lloyd's Classical Result for the Optimum Scalar Quantizer, *IEEE Transactions on Information Theory*, Vol. 32, No. 2, March 1986., pp. 326-328.

[8] N.S. Jayant, Peter Noll, *Digital coding of waveforms*, Prentice-Hall, New Jersey, 1984, Chapter 4, pp. 129-139.

[9] Wai C. Chu, *Speech coding algorithms*, John Wiley & Sons, New Jersey, 2003, Chapter 6, pp. 161-165.